IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) An apparatus, implemented in a computer-readable medium, for subdividing input data associated with a software program and processing each subdivided input data on one or more processing elements, comprising:

a non-threaded initiating program;

one or more non-threaded processing programs, wherein each of the one or more non-threaded processing programs are substantially identical and perform substantially same functions as remaining ones of the one or more threaded processing programs; and

a wrapper that intercepts a call to the initiating program and operable to subdivide input parameters into one or more job quanta, wherein each job quantum is submitted for execution to a separate processing program selected from the one or more processing programs residing on a separate processing element.

2. (Original) The apparatus of claim 1, wherein the wrapper assembles one or more output data from each processing program to form a single results data.

3. (Original) The apparatus of claim 1, wherein each job quantum is provided to a separate job scheduler residing on each of the processing elements, each scheduler manages the execution of the processing program executing on the processing element.

4. (Original) The apparatus of claim 1, further comprising:

one or more additional wrappers, each additional wrapper residing on a single processing element and is operable to intercept the job quantum submitted to the processing program residing on the processing element.

5. (Original) The apparatus of claim 1, wherein the initiating program and each of the processing programs perform one or more operations that are substantially identical.

6. (Original) The apparatus of claim 5, wherein the operations are bioinformatic calculations.

7. (Canceled).

8. (Canceled).

9. (Original) The apparatus of claim 1, wherein at least one of the processing elements resides in a disparate processing environment from the initiating program.

10. (Original) The apparatus of claim 1, wherein the input parameters are normalized prior to being subdivided into the job quanta.

11. (Currently Amended) A method of processing a non-threaded set of executable instructions, comprising:

receiving input data associated with a call to a first non-threaded set of executable instructions;

parsing the input data into a plurality of job quanta, each job quantum operable to be independently processed by the first non-threaded set of executable instructions; and

submitting at least one job quantum for execution to a second non-threaded set of executable instructions, wherein the second non-threaded set of executable instructions ~~that~~ is substantially identical to the first non-threaded set of executable instructions and performs substantially same functions as the first non-threaded set of executable instructions, wherein the second set of executable instructions resides on one or more different processing elements from the first non-threaded set of executable instructions.

12. (Original) The method of claim 11, further comprising:

assembling an output data associated with the results of the execution of the second non-threaded set of executable instructions for a presentation.

13. (Original) The method of claim 11, further comprising:

submitting at least one job quantum for execution to the first non-threaded set of executable instructions.

14. (Original) The method of claim 13, wherein the executions occur substantially in parallel.

15. (Currently Amended) A job quanta data structure, comprising:

a first data;

a second data wherein the first and second data are operable to be delineated and independently submitted as input parameter data used for execution by separate non-threaded sets of executable instructions and processed substantially in parallel on different processing elements, wherein each separate non-threaded set of executable instructions is substantially identical and performs substantially same functions as remaining ones of the non-threaded sets of executable instructions.

16. (Original) The job quanta of claim 15, wherein the first and second data are delineated using extensible markup language.

17. (Original) The job quanta of claim 15, wherein the first and second data are initially submitted as input parameter data to a single non-threaded set of executable instructions.

18. (Currently Amended) A system, implemented in a computer-readable medium, for performing parallel processing on a call to execute a software program, comprising:

means for intercepting a call to the software program, which is non-treaded;

means for dividing a set of input data into a plurality of job quanta including a first job quantum and a second job quantum;

means for submitting the first job quantum to the software program and for submitting the second job quantum to a separate software program, wherein the software program and the

separate software program are substantial identical to one another and perform substantially same functions as one another; and

means for executing the software program and the separate software programs substantially in parallel.

19. (Original) The system of claim 18, further comprising:

means for assembling output data associated with the execution of the software program and at least one of the separate software programs into a presentation data.

20. (Original) The system of claim 19, further comprising:

means for trapping and reporting error conditions generated by the execution of the software program and at least one of the separate software programs.

21. (Currently Amended) A method of processing a software program, comprising:

receiving input data associated with a call to the software program, which is non-threaded;

parsing the input data into a plurality of job quanta, each job quantum operable to be independently processed by the software program; and

submitting at least one job quantum for execution to a replica software program that is substantially identical to the software program and which performs substantially same functions as the software program, wherein the replica software program resides on one or more different processing elements from the software program.

22. (Currently Amended) An information handling system, comprising:

a network;

a plurality of processing elements;

memory operatively coupled to the processing elements; and

means for wrapping a call to a non-threaded application program by dividing input data among the processing elements for execution according to the non-threaded application program and recombining output data from the processing elements, wherein each processing element

includes a duplicate instance of a same non-threaded application and wherein duplicate instance of the same non-threaded application receives a different portion of the divided input data .

23. (Currently Amended) A method of processing a set of executable instructions, comprising:

receiving input data associated with a call to the set of executable instructions, which are non-threaded;

separating the input data into a plurality of job quanta, wherein each job quantum is operable to be independently processed by the set of executable instructions; and

submitting at least one job quantum for execution to a substantial copy of the set of executable instructions, wherein the substantial copy performs same functions as the set of executable instructions, and submitting a different job quantum for execution to the set of executable instructions, wherein the substantial copy of the set of executable instructions and the set of executable instructions reside on different processing elements.

24. (Previously Presented) The method of claim 23, further comprising assembling output data from the execution of the substantial copy of the set of executable instructions and from the set of executable instructions into a single presentation data.

25. (Previously Presented) The method of claim 23, further comprising executing the substantial copy of the set of executable instructions and the set of executable instructions substantially in parallel.

26. (Previously Presented) The method of claim 23, wherein in separating the input data, the input data is separated into the plurality of job quanta by a wrapper associated with the set of executable instructions.

27. (Currently Amended) A parallel processing system, implemented in a computer-readable medium, comprising:

a first software program having a wrapper operable to intercept calls made to the first software program, wherein the first software program resides on one or more first processing elements and is non-threaded; and

a second software program which is a substantial copy of the first software program, which performs substantially same functions as the first software program, and which is non-threaded, wherein the second software program resides on one or more second processing elements, and

wherein the wrapper intercepts the calls and parses input data associated with the calls into job quanta, the job quanta includes a first job quantum and a second job quantum, and the first job quantum is submitted to the first software program for processing and the second job quantum is submitted to the second software program for processing substantially in parallel.

28. (Previously Presented) The system of claim 27, wherein the wrapper assembles output results associated with the processing of the first job quantum and the second job quantum.

29. (Previously Presented) The system of claim 27, wherein one or more of the first processing elements are different from one or more of the second processing elements.

30. (Currently Amended) A parallel processing system, comprising:

a wrapper that intercepts calls to non-threaded software programs, wherein the software programs are substantial copies of each other and perform same functions as each other and which reside on different processing elements, and wherein the wrapper separates input data associated with the calls into a plurality of independent job quanta; and

a scheduler that receives the plurality of job quanta from the wrapper and submits substantially in parallel different job quantum associated with the job quanta to a number of the software programs for processing, wherein the scheduler selects the number of the software programs based on processing loads associated with the number of the software programs.

31. (Previously Presented) The system of claim 30, wherein the wrapper assembles results associated with processing the different job quantum for a unified presentation.

32. (Previously Presented) The system of claim 30, wherein the scheduler traps any errors associated with processing the different job quantum and reports the errors to the wrapper.